HL Klemove

# Trajectory planning

Lecturer : Seungmok Song

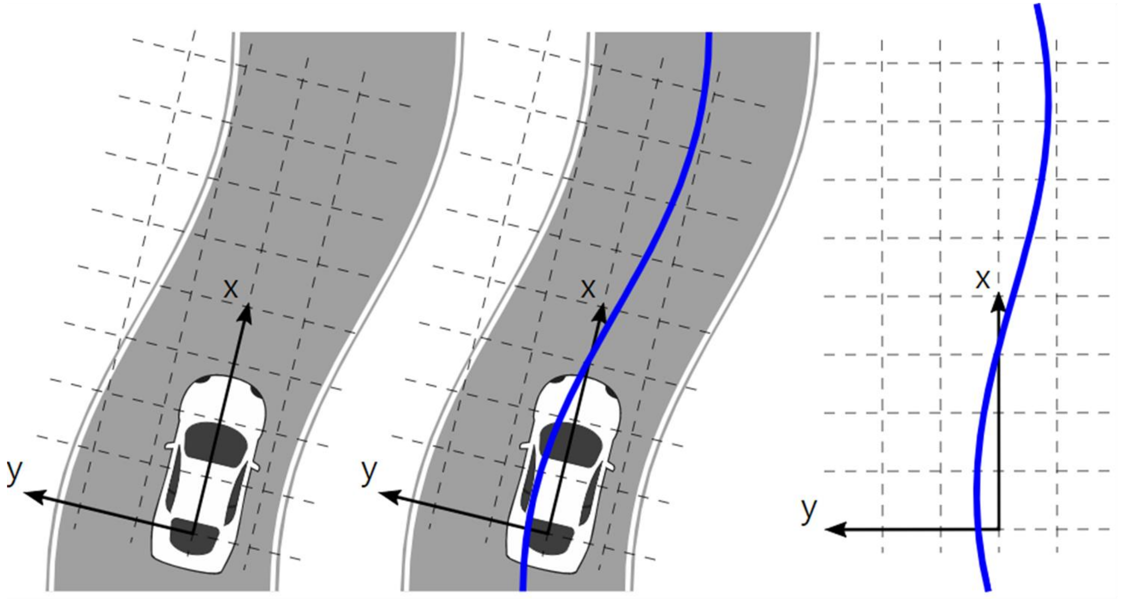
# Contents

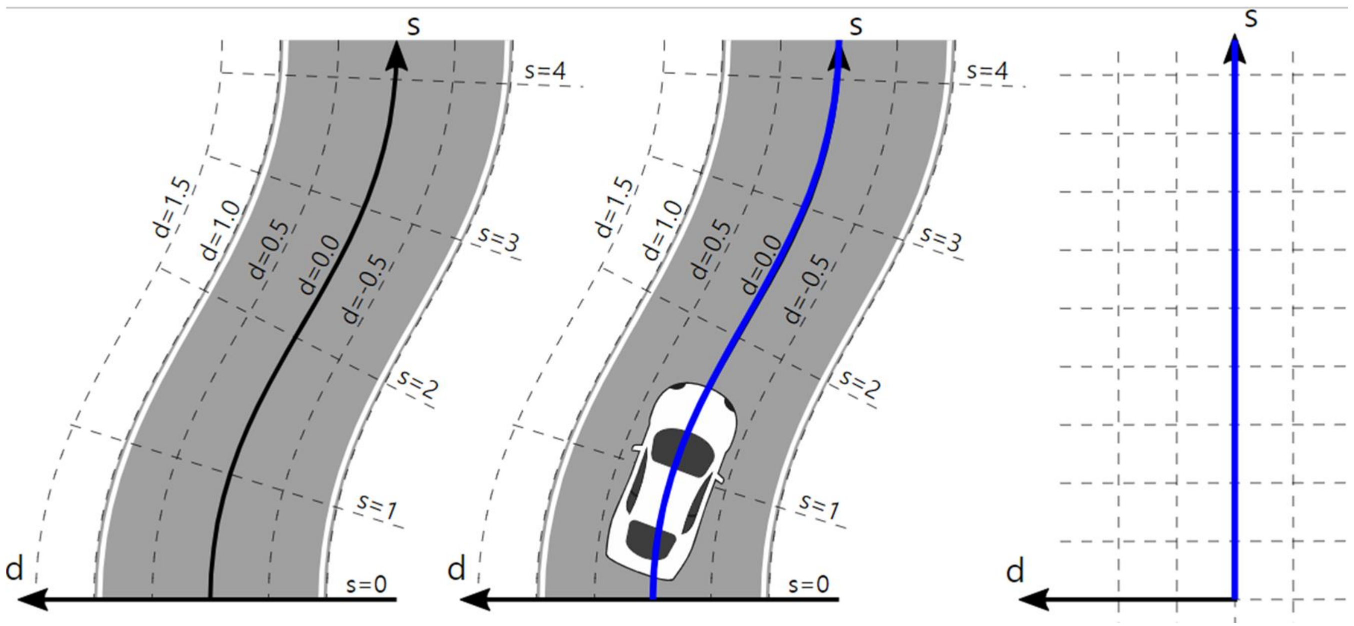
# Introduction

- Trajectory planning
  - Why do we need trajectory?

# Frenet frame

- Concept of Frenet frame
  - Figure (a) 은 Cartesian frame 을, Figure (b) 은 Frenet frame 을 표현
  - 도로 중심(path) 방향으로 s –axis 가, 도로와 수직방향으로 d-axis 가 펼쳐짐
  - 따라서 Frenet frame 을 이용하면 planning 관점에서 더 직관적이고 편리함






Source: https://fjp.at/posts/optimal-frenet/

# Frenet frame

- Concept of Frenet frame

**2010 IEEE International Conference on Robotics and Automation**
**Anchorage Convention District**
**May 3-8, 2010, Anchorage, Alaska, USA**

## Optimal Trajectory Generation for Dynamic Street Scenarios in a Frenét Frame

Moritz Werling, Julius Ziegler, Sören Kammel, and Sebastian Thrun

***Abstract*— Safe handling of dynamic highway and inner city scenarios with autonomous vehicles involves the problem of generating traffic-adapted trajectories. In order to account for the practical requirements of the holistic autonomous system, we propose a semi-reactive trajectory generation method, which can be tightly integrated into the behavioral layer. The method realizes long-term objectives such as velocity keeping, merging, following, stopping, in combination with a reactive collision avoidance by means of optimal-control strategies within the Frenét-Frame [12] of the street. The capabilities of this approach are demonstrated in the simulation of a typical high-speed highway scenario.**

### I. Introduction

#### A. Motivation

The past three decades have witnessed ambitious research

to the planning level. Additionally, the algorithm provides for reactive obstacle avoidance by the *combined* usage of steering and breaking/acceleration.

#### B. Related work

Several methods for trajectory planning have been proposed [11], [19], [2], [4] that find a global trajectory connecting a start and a - possibly distant - goal state. However, these methods fail to model the inherent unpredictability of other traffic, and the resulting uncertainty, given that they rely on precise prediction of other traffic participant's motions over a long time period. Other approaches taken towards trajectory planning follow a discrete optimization scheme (e. g. [16], [1], [7]): A finite set of trajectories is computed, typically by forward integration of the differential

as during initialization and lane changes, or when we switch between low and high speed trajectories, we have to project the current end point $(x, \theta_x, \kappa_x, v_x, a_x)(t_0)$ on the new center line and determine the corresponding $[s_0, \dot{s}_0, \ddot{s}_0, d_0, \dot{d}_0, \ddot{d}_0]$ or $[s_0, \dot{s}_0, \ddot{s}_0, d_0, d'_0, d''_0]$ respectively. For this reason, the transformations in the appendix can easily be inverted in closed form, except for $s_0$, as we do not restrict the center line $\vec{r}(s)$ to a certain shape[6]. However the inversion can be restated as the minimization problem $s = \underset{\sigma}{\operatorname{argmin}} \|x - r(\sigma)\|$, for which efficient numerical methods exist.

# Frenet frame

- Concept of Frenet frame

APPENDIX I
TRANSFORMATIONS FROM FRENÉT COORDINATES TO GLOBAL COORDINATES

In addition to (1), we seek for transformations

$$[s, \dot{s}, \ddot{s}; d, \dot{d}, \ddot{d}/d, d', d''] \mapsto [\vec{x}, \theta_x, \kappa_x, v_x, a_x]$$

The mayor challenges at this is to handle the singularity $v_x = 0$. Therefore we introduce $\vec{t}_r(s) := \begin{bmatrix} \cos\theta_r(s) & \sin\theta_r(s) \end{bmatrix}^{\mathrm{T}}$ and $\vec{n}_r(s) := \begin{bmatrix} -\sin\theta_r(s) & \cos\theta_r(s) \end{bmatrix}^{\mathrm{T}}$, where $\theta_r(s)$, $\vec{t}_r(s)$ and $\vec{n}_r(s)$ are the orientation, the tangential and normal vectors of the center line in $s$. In addition, we denote the curvature as $\kappa_r$ and assume that we travel along the center line excluding extreme situations, such that $\|\Delta\theta\|_2 < \frac{\pi}{2}$, with $\Delta\theta := \theta_x - \theta_r$, and $1 - \kappa_r d > 0$ at all times. As we can derive the transformation needed for higher speeds from the one associated with lower speeds, we will start with the latter. With (1) we get

$$d = [\vec{x} - \vec{r}(s)]^{\mathrm{T}} \vec{n}_r. \tag{4}$$

Time derivative yields with $\dot{\vec{n}}_r = -\kappa_r \vec{t}_r$

$$\begin{aligned} \dot{d} &= [\dot{\vec{x}} - \dot{\vec{r}}(s)]^{\mathrm{T}} \vec{n}_r + [\vec{x} - \vec{r}(s)]^{\mathrm{T}} \dot{\vec{n}}_r \\ &= v_x \vec{t}_x^{\,\mathrm{T}} \vec{n}_r - \dot{s}\, \underbrace{\vec{t}_r^{\,\mathrm{T}} \vec{n}_r}_{=0} - \kappa_r \underbrace{[\vec{x} - \vec{r}(s)]^{\mathrm{T}} \vec{t}_r}_{=0} = v_x \sin\Delta\theta. \end{aligned} \tag{5}$$

Therefore, we calculate

$$\begin{aligned} v_x = \|\dot{x}\|_2 &= \left\| \begin{bmatrix} \vec{t}_r & \vec{n}_r \end{bmatrix} \begin{bmatrix} 1 - \kappa_r d & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{s} \\ \dot{d} \end{bmatrix} \right\|_2 \\ &= \left\| \begin{bmatrix} 1 - \kappa_r d & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{s} \\ \dot{d} \end{bmatrix} \right\|_2 = \sqrt{[1 - \kappa_r d]^2 \dot{s}^2 + \dot{d}^2} \end{aligned}$$

$$\begin{aligned} \text{and} \quad d' &:= \frac{\mathrm{d}}{\mathrm{d}s} d = \frac{\mathrm{d}t}{\mathrm{d}s} \frac{\mathrm{d}}{\mathrm{d}t} d = \frac{1}{\dot{s}} \dot{d} = \frac{1}{\dot{s}} v_x \sin\Delta\theta \\ &= \sqrt{[1 - \kappa_r d]^2 + d'^2} \sin\Delta\theta \\ d'^2 &= [[1 - \kappa_r d]^2 + d'^2] \sin^2\Delta\theta \\ d'^2 [1 - \sin^2\Delta\theta] &= [1 - \kappa_r d]^2 \sin^2\Delta\theta, \end{aligned}$$

so that we get $\quad d' = [1 - \kappa_r d] \tan\Delta\theta. \qquad (6)$

Additionally, we know that $[\vec{x} - \vec{r}]^{\mathrm{T}} \vec{t}_r = 0$ at all times, so that differentiating with respect to time gives us analog to (5) $\frac{v_x}{\dot{s}} \cos\Delta\theta - 1 + \kappa_r d = 0$ and we can solve for the velocity

$$v_x = \dot{s} \frac{1 - \kappa_r d}{\cos\Delta\theta}. \tag{7}$$

With this and $s_x$ being the arc length of the trajectory $\vec{x}$, we can conclude that

$$\frac{\mathrm{d}}{\mathrm{d}s} = \frac{\mathrm{d}s_x}{\mathrm{d}s} \frac{\mathrm{d}}{\mathrm{d}s_x} = \frac{\mathrm{d}s_x}{\mathrm{d}t} \frac{\mathrm{d}t}{\mathrm{d}s} \frac{\mathrm{d}}{\mathrm{d}s_x} = \frac{v_x}{\dot{s}} \frac{\mathrm{d}}{\mathrm{d}s_x} = \frac{1 - \kappa_r d}{\cos\Delta\theta} \frac{\mathrm{d}}{\mathrm{d}s_x}, \tag{8}$$

so that we calculate the second derivative of $d$ to be

$$\begin{aligned} d'' &= -[\kappa_r d]' \tan\Delta\theta + \frac{1 - \kappa_r d}{\cos^2\Delta\theta} \left[ \frac{\mathrm{d}}{\mathrm{d}s} \theta_x - \theta_r' \right] \\ &= -[\kappa_r' d + \kappa_r d'] \tan\Delta\theta + \frac{1 - \kappa_r d}{\cos^2\Delta\theta} \left[ \kappa_x \frac{1 - \kappa_r d}{\cos\Delta\theta} - \kappa_r \right]. \end{aligned} \tag{9}$$

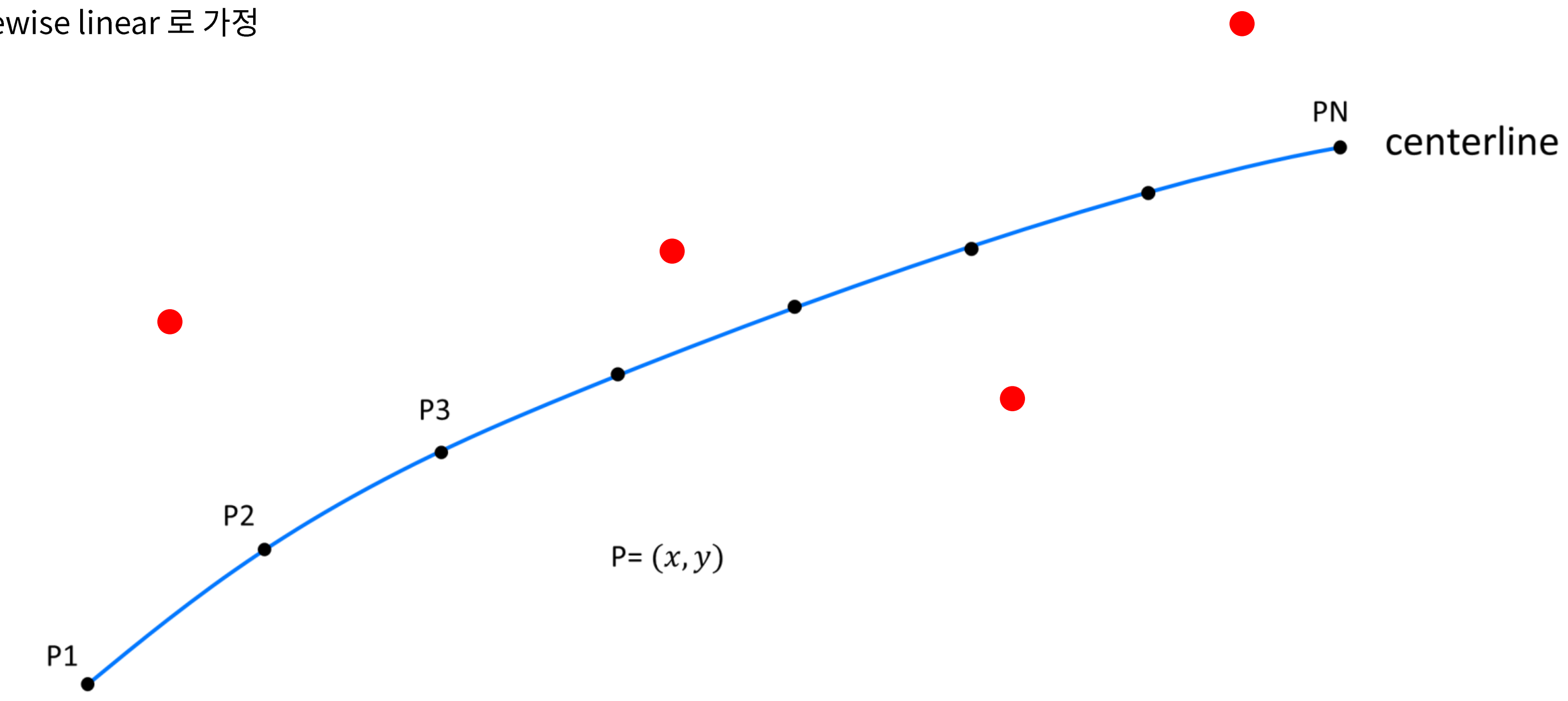
Equations (6) and (9) can be solved for $\theta_x$ and $\kappa_x$, including $v_x = 0$. Time differentiating the velocity once more yields the last unknown in our transformation

$$\begin{aligned} a_x &:= \dot{v}_x = \ddot{s} \frac{1 - \kappa_r d}{\cos\Delta\theta} + \dot{s} \frac{\mathrm{d}}{\mathrm{d}s} \frac{1 - \kappa_r d}{\cos\Delta\theta} \dot{s} = \\ &\ddot{s} \frac{1 - \kappa_r d}{\cos\Delta\theta} + \frac{\dot{s}^2}{\cos\Delta\theta} \left[ [1 - \kappa_r d] \tan\Delta\theta \Delta\theta' - [\kappa_r' d + \kappa_r d'] \right]. \end{aligned}$$

For high speed driving we calculate $\dot{d} = \frac{\mathrm{d}}{\mathrm{d}t} d = \frac{\mathrm{d}s}{\mathrm{d}t} \frac{\mathrm{d}}{\mathrm{d}s} d = \dot{s}\, d$ and $\ddot{d} = d'' \dot{s}^2 + d' \ddot{s}$. As $\dot{s} \neq 0$ holds for higher speeds, subsequently solving these equations for $d'$ and $d''$ enables us to use the previously calculated transformations. Notice, that the center line $\vec{r}(s)$ needs to have a continuous change of curvature $\kappa_r'$ in order to provide for a trajectory $\vec{x}(t)$ with a continuous $\kappa_x$.

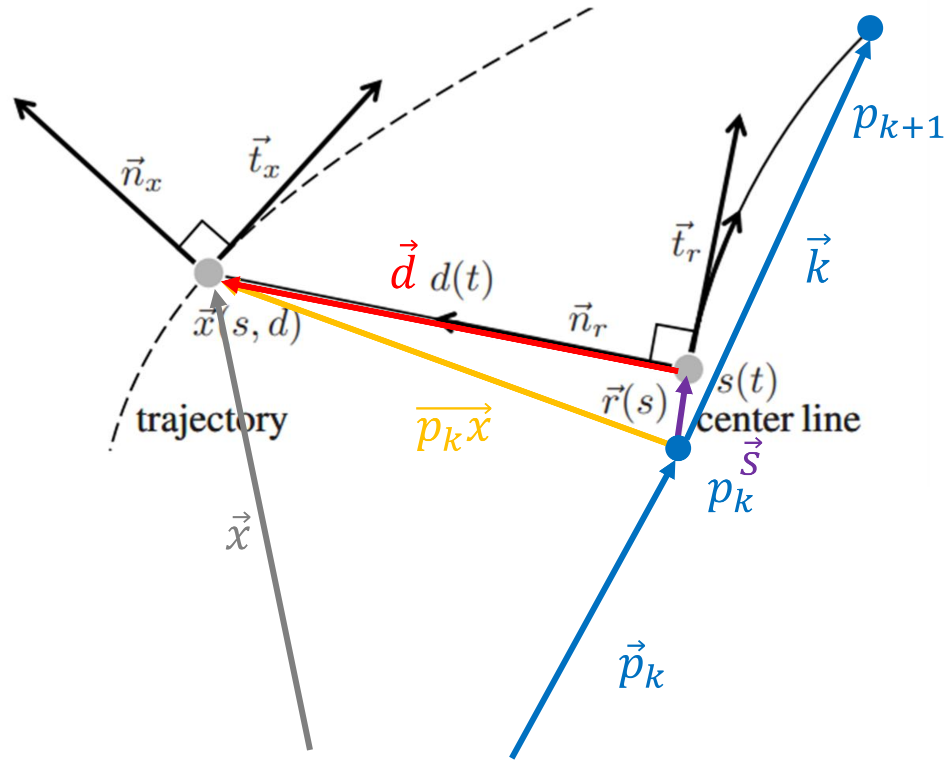
# Frenet frame

- Cartesian → Frenet frame(simple way)
  - What we have
    - Centerline = map의 (x, y) 좌표
    - x 좌표 → mapx = [x1, x2, ⋯, xn]
    - y 좌표 → mapy = [y1, y2, ⋯, yn]
    - Centerline : Piecewise linear 로 가정



# Frenet frame

- Cartesian → Frenet frame conversion
  - Main idea 1
    - Centerline 위에서 현재 (x, y)와 가장 가까운 점 찾기
    - Vector 들을 이용



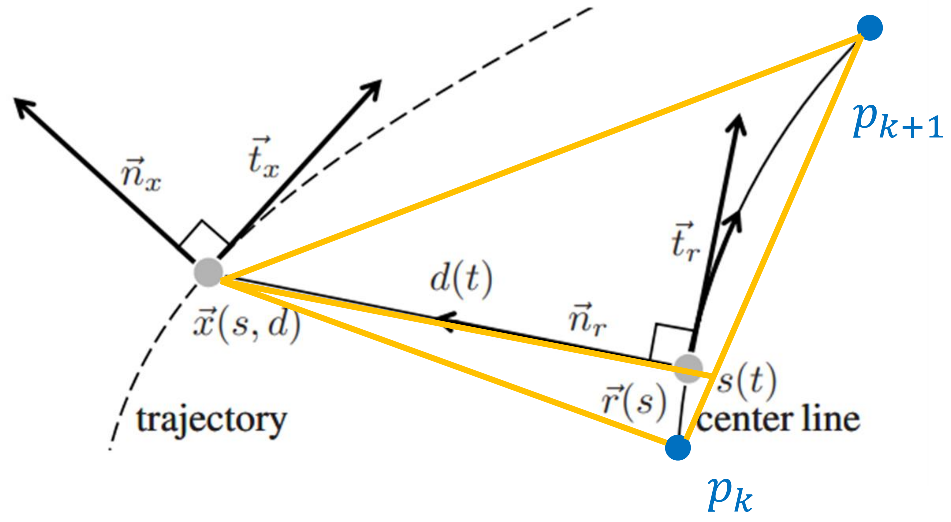

$$\vec{d} = \vec{x} - (\vec{p}_k + \vec{s})$$

$$\vec{s} = proj(\overrightarrow{p_k x}, \vec{k}) = \left(\frac{\overrightarrow{p_k x} \cdot \vec{k}}{\vec{k} \cdot \vec{k}}\right)\vec{k}$$

HL Klemove

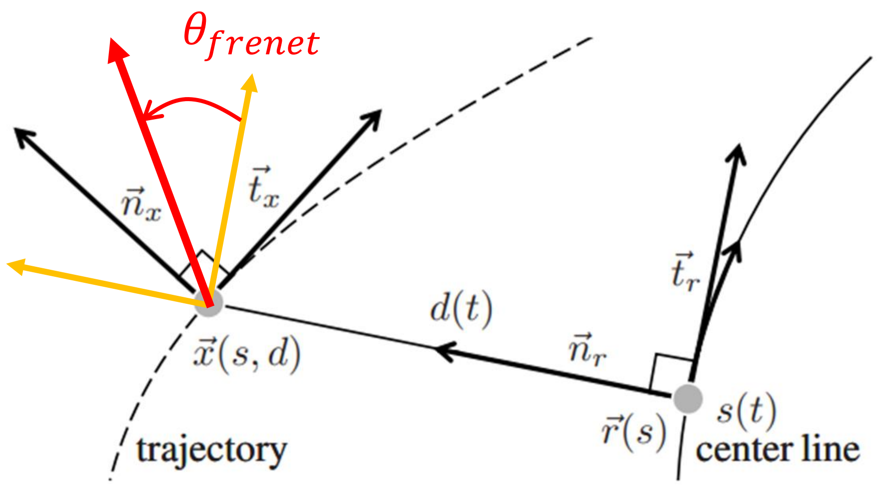
# Frenet frame

- Cartesian → Frenet frame conversion
  - Main idea 2
    - Centerline 위에서 현재 (x, y)와 가장 가까운 점 찾기
    - Sine 법칙 이용(부호만 조심)



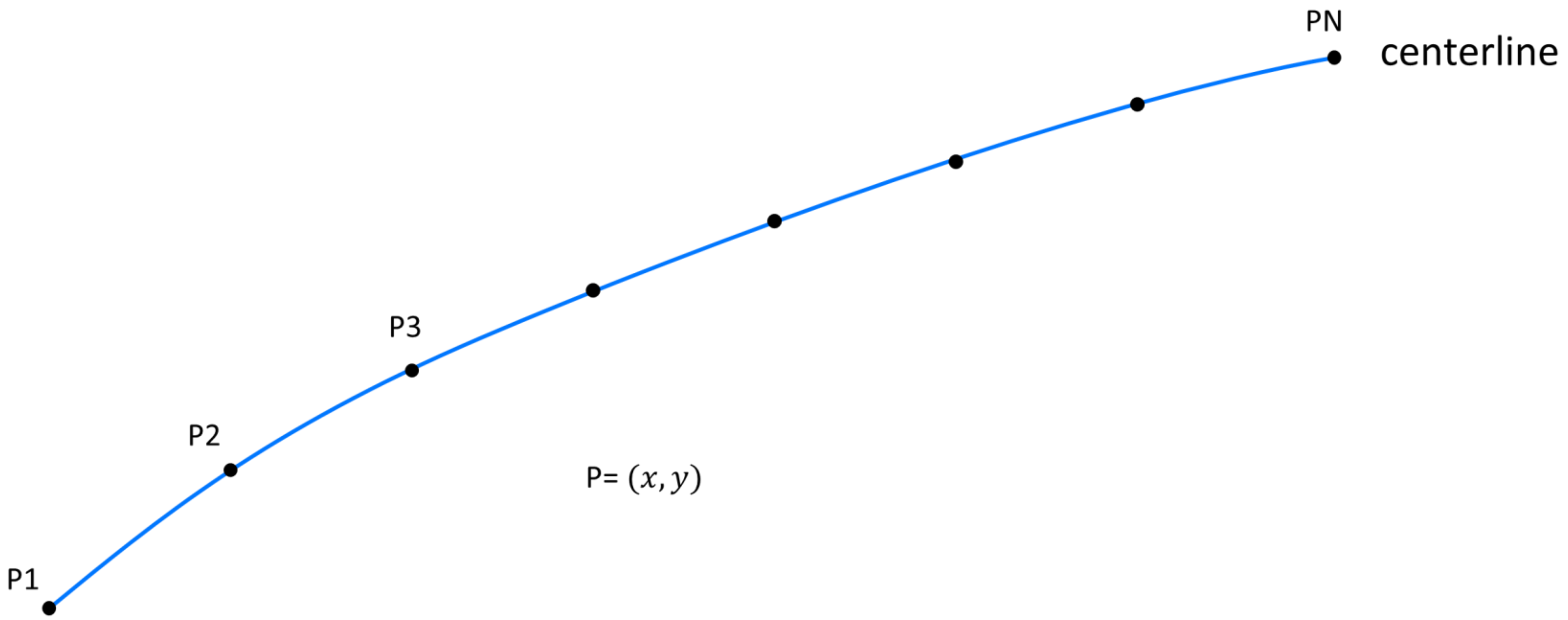
# Frenet frame

- Cartesian → Frenet frame conversion
  - Vector conversion
    - 방향성 있는 물리량의 변환 : velocity, acceleration, …



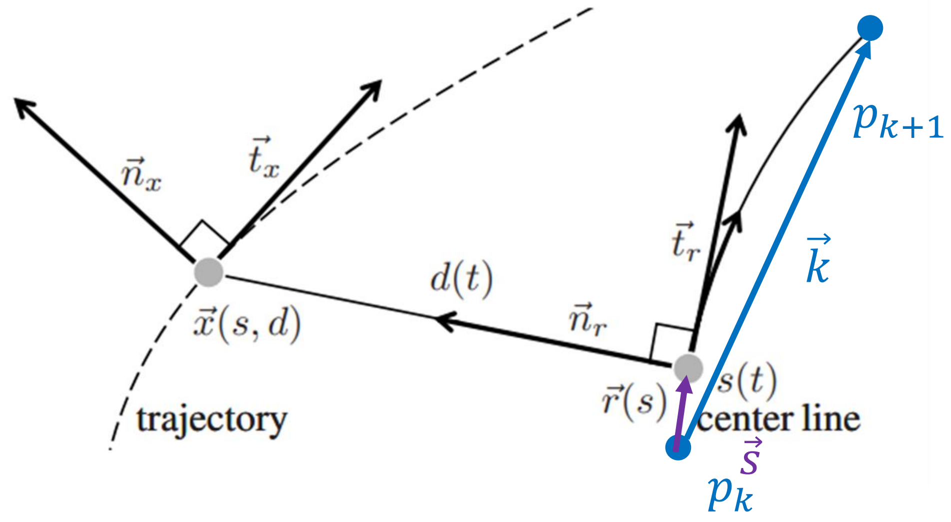
# Frenet frame

- Frenet → Cartesian frame(simple way)
  - What we have
    - Centerline = map의 (x, y) 좌표
    - s 좌표 → maps = [s1, s2, ⋯, sn]
    - d 좌표 → mapd = [d1, d2, ⋯, dn]
    - Centerline : Piecewise linear 로 가정



# Frenet frame

- Frenet → Cartesian frame(simple way)
  - Main idea
    - Pk 찾기 : 쉬움 (s,d) 를 알기때문에




$$s - (l_0 + l_1 + \cdots + l_k) = s_k$$

$$\vec{s} = s_k \cdot \frac{\vec{k}}{|\vec{k}|}$$

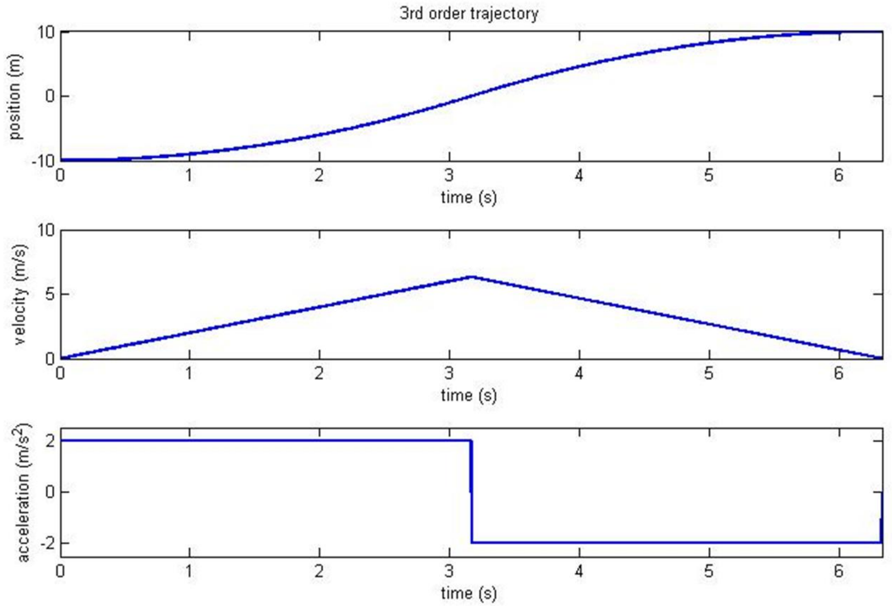
$$\vec{k} \perp \vec{d}$$

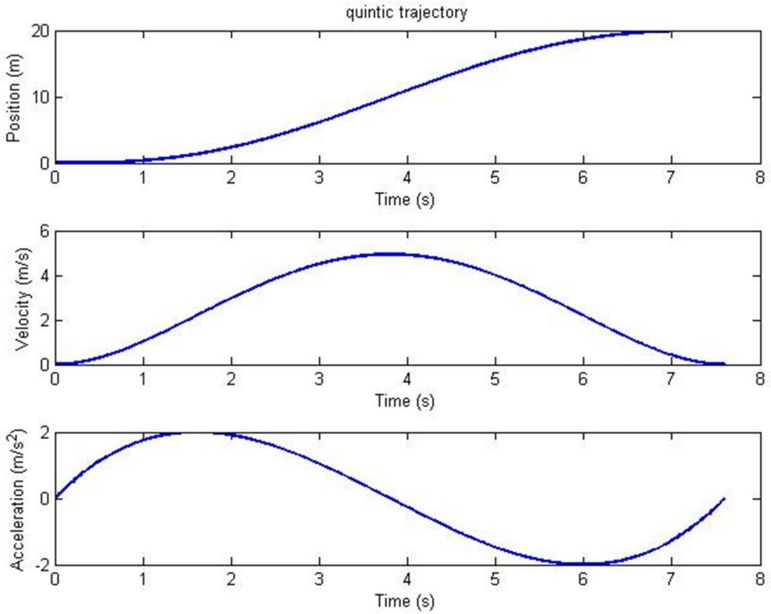
$$\vec{x} = \vec{p}_k + \vec{s} + \vec{d}$$

# Optimal trajectory planning

- Polynomial trajectory generation
  - Widely used!

$$d(t) = {n_d}^{th} polynomial$$
$$n(t) = {n_n}^{th} polynomial$$

# Optimal trajectory planning

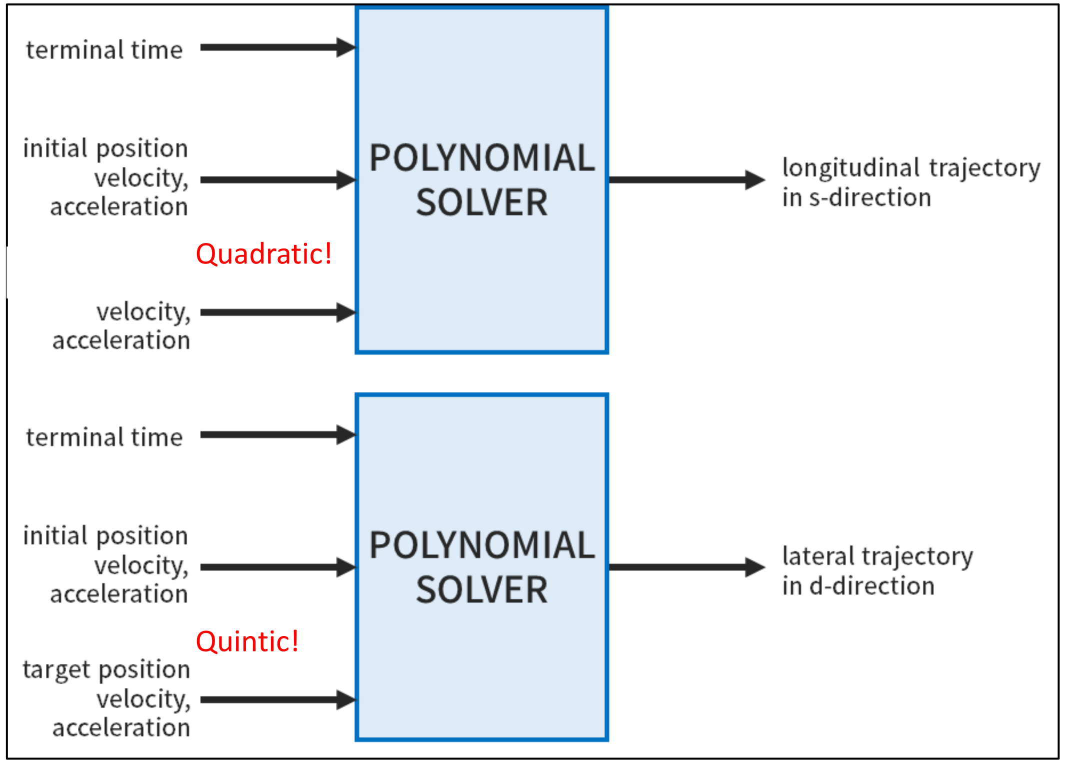
- Polynomial trajectory generation
  - 종 횡방향 각각 경로 생성
    - Longitudinal trajectory : quartic (4차) or quintic(5차) polynomial
    - Lateral trajectory : quintic (5차) polynomial

Position $q(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5$

Velocity $\dot{q}(t) = a_1 + 2a_2 t + 3a_3 t^2 + 4a_4 t^3 + 5a_5 t^4$

Acceleration $\ddot{q}(t) = 2a_2 + 6a_3 t + 12a_4 t^2 + 20a_5 t^3$

Given: $q_0, \dot{q}_0, \ddot{q}_0, q_f, \dot{q}_f, \ddot{q}_f$




$$\begin{bmatrix} 1 & t_i & t_i^2 & t_i^3 & t_i^4 & t_i^5 \\ 0 & 1 & 2t_i & 3t_i^2 & 4t_i^3 & 5t_i^4 \\ 0 & 0 & 2 & 6t_i & 12t_i^2 & 20t_i^3 \\ 1 & t_f & t_f^2 & t_f^3 & t_f^4 & t_f^5 \\ 0 & 1 & 2t_f & 3t_f^2 & 4t_f^3 & 5t_f^4 \\ 0 & 0 & 2 & 6t_f & 12t_f^2 & 20t_f^3 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix} = \begin{bmatrix} q_i \\ \dot{q}_i \\ \ddot{q}_i \\ q_f \\ \dot{q}_f \\ \ddot{q}_f \end{bmatrix}$$

# Optimal trajectory planning

- Polynomial trajectory generation(Candidates)
  - Longitudinal trajectory
    - Following, Merging and Stopping
    - Initial condition (3개) : position, velocity, acceleration
    - Terminal condition (3개) : position, velocity, acceleration

    - Velocity Keeping
    - Initial condition (3개) : position, velocity, acceleration
    - Terminal condition (2개) : velocity, acceleration

  - Lateral trajectory
    - Initial condition (3개) : position, velocity, acceleration
    - Terminal condition (3개) : position, velocity, acceleration
    - Jerk minimizing trajectory

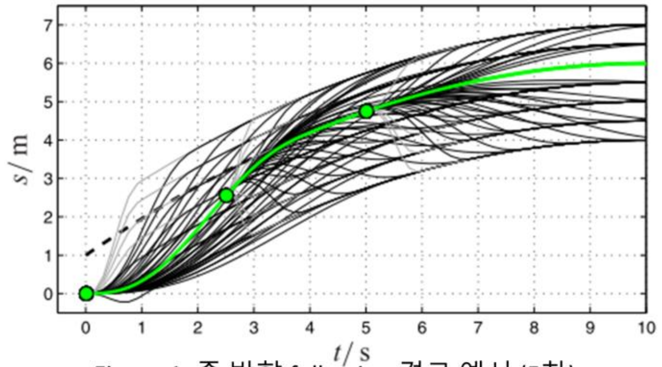


Figure 1. 종 방향 following 경로 예시 (5차)

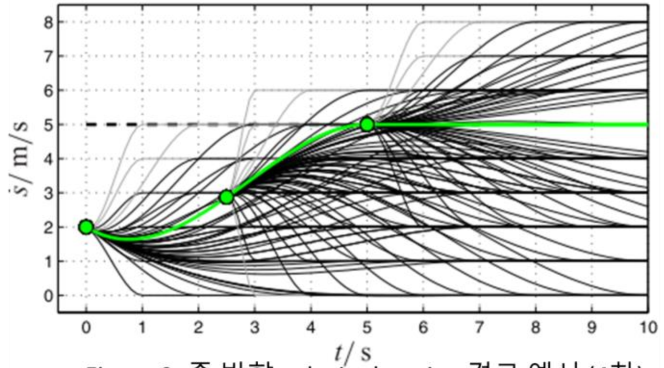


Figure 2. 종 방향 velocity keeping 경로 예시 (4차)

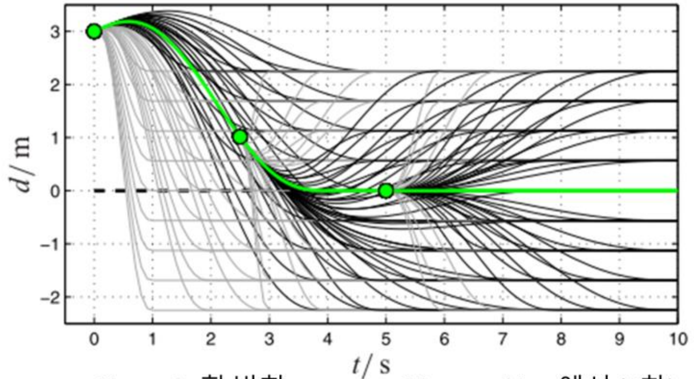


Figure 3. 횡 방향 target position tracking 예시 (5차)

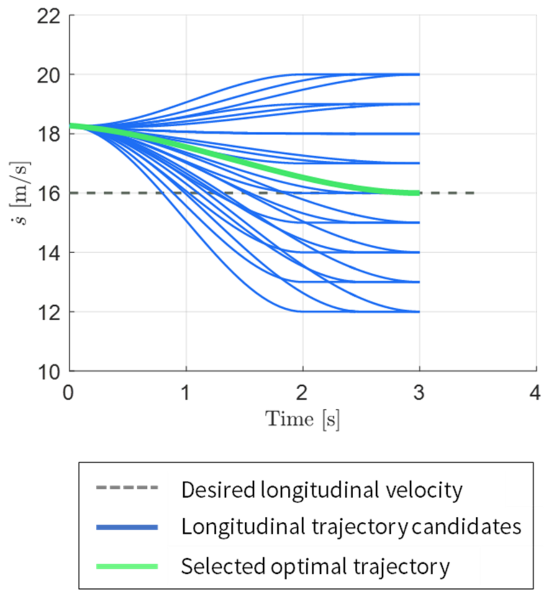
*Refs : Werling et el. Optimal trajectories for time-critical street scenarios using discretized terminal manifolds. 2012 IJRR.

# Optimal trajectory planning

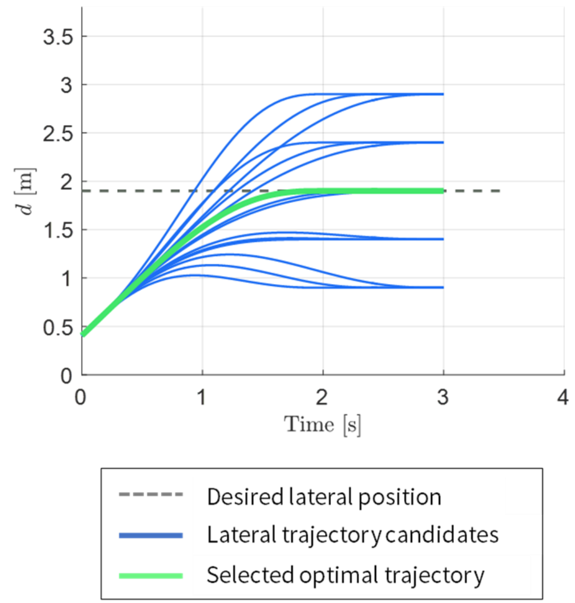
- Combining longitudinal and lateral trajectories
  - 종/횡 방향 경로 통합

$$d(t) = {n_d}^{th} polynomial$$
$$n(t) = {n_n}^{th} polynomial$$

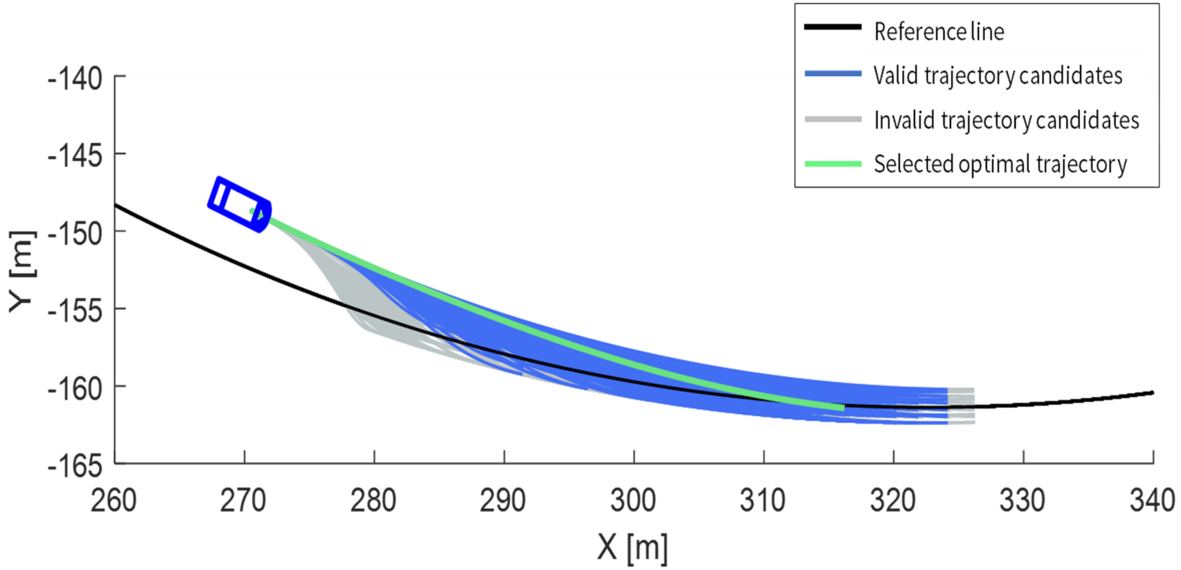
Longitudinal trajectories




Lateral trajectories




Combined trajectories



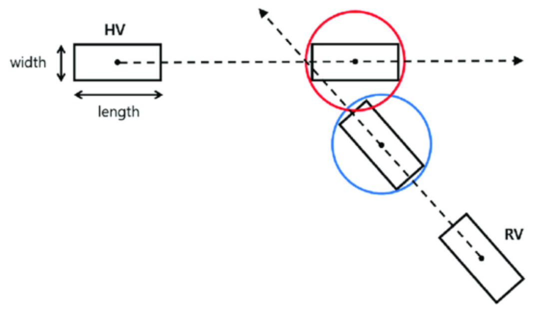
# Optimal trajectory planning

- Trajectory check
  - Constraints
    - Kinematic and dynamic constraint check
    - Collision check

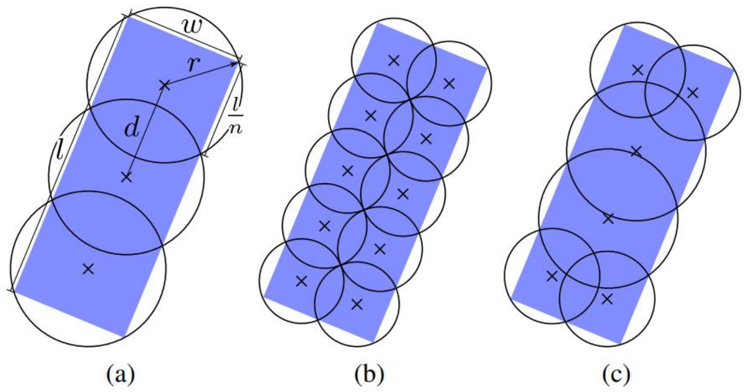
| | | Restrictions |
|---|---|---|
| Velocity | | $\dot{s}(t) \leq \dot{s}_{limit}$ |
| Curvature | - vehicle kinematic check | $\kappa_{cand} \in [\kappa_{min}, \kappa_{max}]$ |
| Acceleration | - vehicle dynamic check | $\ddot{s}(t)^2 + \ddot{d}(t)^2 \leq a_{max}^2$ |
| Collision check | - collision check | Euclidean distance-based check |

# Optimal trajectory planning

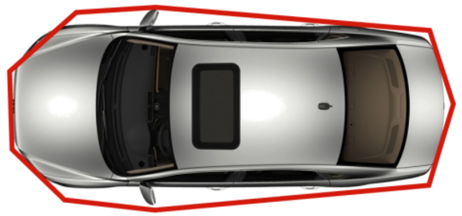
- Trajectory check
  - Approximating check : Approximating as circles

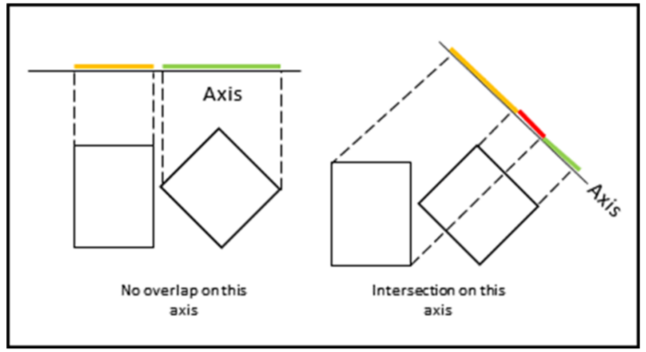


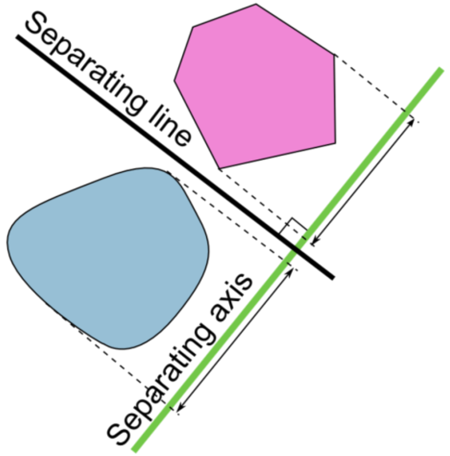

Minjin B., "Vehicle Trajectory Prediction and Collision Warning via Fusion of Multisensors and Wireless Vehicular Communications" (2020)




  - Exact collision check : Separating axis theorem

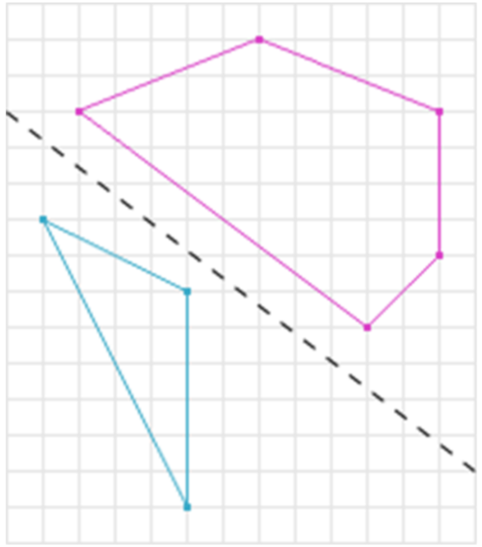




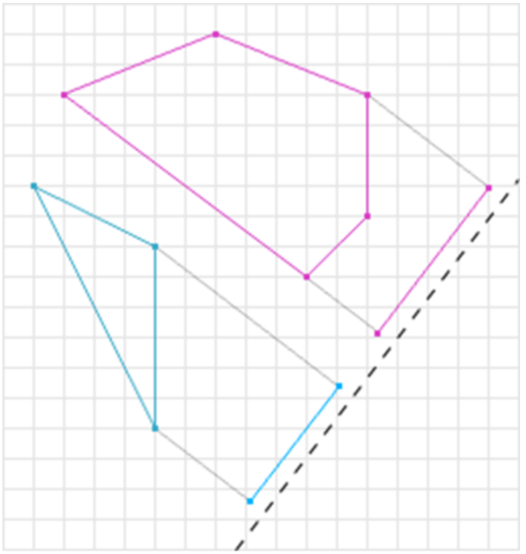



# Optimal trajectory planning

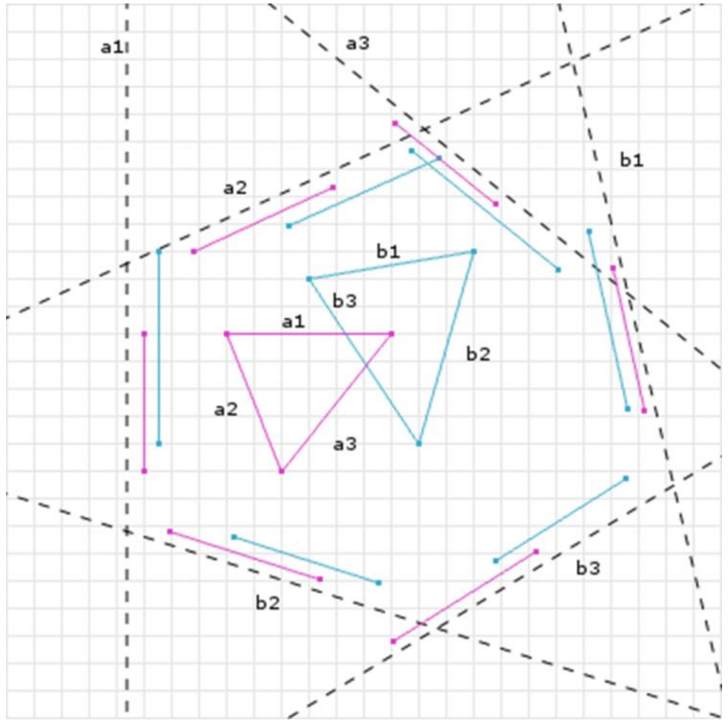
- Trajectory check
  - Separating axis theorem
    - Approximate to convex polygon(if not, need to separate into convex polygons)
    - Rapid & accurate way to check collision
    - Axis : perpendicular to every line



Two Separated Convex Shapes



Two Separated Convex Shapes With Their Respective Projections




Two Convex Shapes Intersecting

# Optimal trajectory planning

- Optimal trajectory selection
  - Cost function에 따른 최적 경로 선택

Minimize , $J_{total} = w_s J_s + w_d J_d$

$$J_s = c_{j,s} \int_{t_i}^{t_f} (\dddot{s}(t))^2 dt + c_{v,s}\left(\dot{s}_{set} - \dot{s}_f\right)^2 + c_{T,s} T$$

longitudinal jerk,
velocity deviation,
time interval

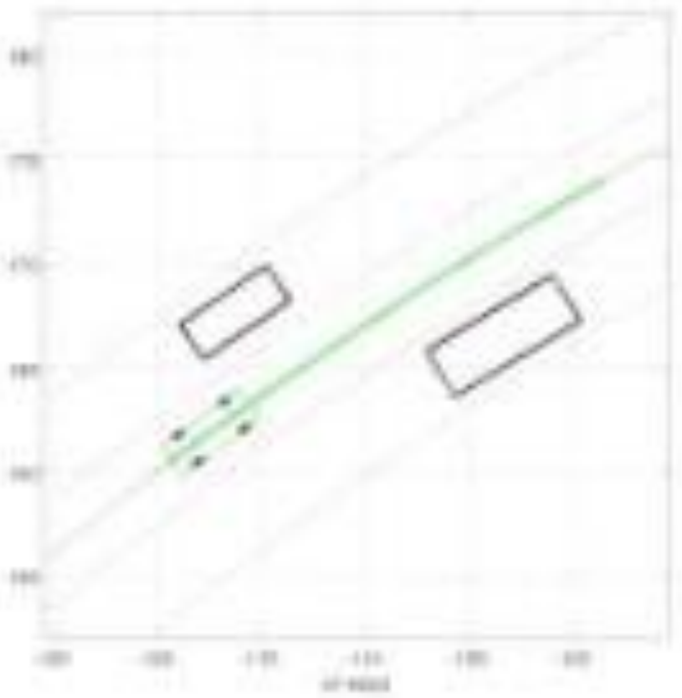
$$J_d = c_{j,d} \int_{t_i}^{t_f} \left(\dddot{d}(t)\right)^2 dt + c_{T,d} T + \left(d_f - d_{f,opt}\right)^2$$

lateral jerk,
time interval,
terminal lateral deviation

이전 optimal 경로와 횡방향 변화를 줄이기 위함 (경로의 떨림 방지)

# Optimal trajectory planning

- Example : Highway scenario demonstration
  - Video: https://youtu.be/Cj6tAQe7UCY
  - Paper: https://www.researchgate.net/publication/224156269_Optimal_Trajectory_Generation_for_Dynamic_Street_Scenarios_in_a_Frenet_Frame



HL Klemove

# Thank You